NPS55GV72091A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

APPROXIMATE MODELS FOR PROCESSOR UTILIZATION

IN MULTIPROGRAMMED COMPUTER SYSTEMS

D. P. Gaver
Naval Postgraduate School

G. S. Shedler
IBM Research Laboratory

September 1972

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M. B. Freeman, USN
Superintendent

M. U. Clauser
Provost

ABSTRACT

This paper presents results of an approximation study of cyclic queueing phenomena that occur in multiprogrammed computer systems. Based on Wald's Identity and using ideas of diffusion, the objective is to develop convenient and nearly explicit formulas relating processor utilization in such systems to simple program parameters and the level of multiprogramming. Some numerical results to indicate the quality of the proposed approximation are given.

Prepared by:

APPROXIMATE MODELS FOR PROCESSOR UTILIZATION IN MULTIPROGRAMMED COMPUTER SYSTEMS

D. P. Gaver*
Naval Postgraduate School
Monterey, California

G. S. Shedler
IBM Research Laboratory
San Jose, California

ABSTRACT.

This paper presents results of an approximation study of cyclic queueing phenomena that occur in multiprogrammed computer systems. Based on Wald's Identity and using ideas of diffusion, the objective is to develop convenient and nearly explicit formulas relating processor utilization in such systems to simple program parameters and the level of multiprogramming. Some numerical results to indicate the quality of the proposed approximation are given.

1. Introduction.

In a previous paper [2] we have initiated an approximation study of cyclic queueing phenomena that occur in multiprogrammed computer systems. Particular attention was focused upon processor utilization estimation, as the latter depends upon the statistical properties of programs. The basis for the approximation was the observation that under "heavy traffic" conditions it is plausible to approximate the flow of programs in a multiprogrammed computer system by means of a diffusion or Wiener process with appropriate infinitesimal

*This author is also a consultant to IBM Research.

parameters and boundary conditions. The results were seen to be usefully accurate, as judged numerically, and to be of an extremely simple analytical form. They can thus be put to use for at least preliminary design purposes, with follow-up refined analysis or simulation furnishing further corrections if needed.

One deficiency of the results of [2] is that they tend to misestimate CPU utilization (i.e. the long-run fraction of time that the CPU is busy) when CPU service or processing times come from distributions of greater positive skewness than the exponential. In the present paper we wish to alter our approximation so as to render it more accurate in the case of such hyper-exponential-appearing CPU service times. This change is important, since currently available data indicates that greater-than-exponential skewness is not uncommon.

2. The Model.

We suppose, as we did in [2], that J programs are in the Central Processing Unit (CPU) - Data Transfer Unit (DTU) cycle. Each program is (i) in the process of awaiting, or receiving, service at the CPU, at the termination of which (ii) it repairs to the DTU, again queueing as if at a single server. Having received the requisite information at the DTU stage it then returns to the CPU stage. This process continues indefinitely. When programs are completed and thus removed from the system new programs are immediately reintroduced. A diagram indicating the situation appears below.

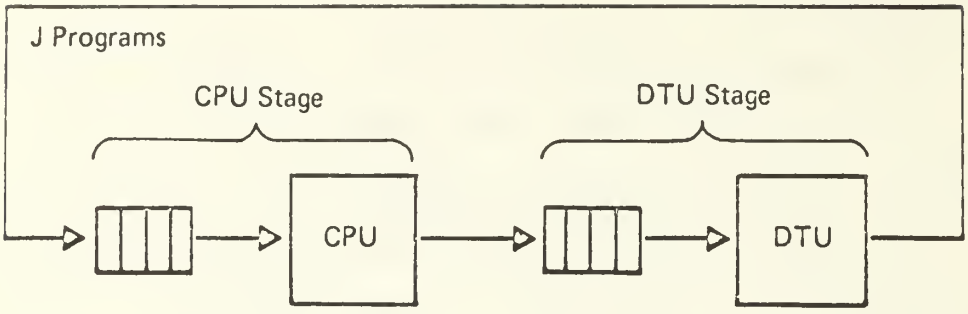
The assumptions made concerning program behavior are the following:

(a) The sequence of CPU service or processing times is one of independent identically distributed random variables (i.i.d.r.v.) $\{C_i, \ i = 1,2,...\}$.

(b) The sequence of DTU service or auxiliary memory access and data transfer times is also one of i.i.d.r.v., $\{D_i\}$.

(c) CPU and DTU processing times are mutually independent. Furthermore, we must assume the following.

(d) The Laplace transform, $E[e^{-sC}]$, of a generic CPU service time converges for $-s_0 < s < 0$, for some $s_0 > 0$. This latter is truly a mathematical restriction, but is probably not a serious one; all gamma densities, and convex combinations of exponentials (hyperexponentials) are covered, for example.


J Programs
CPU Stage
DTU Stage
CPU
DTU

3. Analysis of the Model.

In summary, our present approximate analysis of the multiprogramming model proceeds by first attempting to find an appropriate set of parameters μ and σ in the diffusion equation

$$\frac{\partial F}{\partial t} = - \mu \frac{\partial F}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 F}{\partial x^2} . \tag{3.1}$$

Here $F(x,t)$ is the approximate distribution of the number of jobs in the CPU stage at time t. An argument for obtaining parameters μ and σ^2 which was based on asymptotic renewal theory appears in [2]. Then we truncate the stationary distribution to allow for the boundary at $x = J$, and compare several methods for determining a crucial constant (denoted by B in [2]) that allows us to deal with the boundary at $x = 0$. The latter is important for it is directly related to CPU utilization, which it is our intention to estimate.

Consider the waiting time, S_n, of the n^{th} customer to arrive at an ordinary single server--one in which there is no restriction placed upon the number waiting. The latter model would approximate the behavior of a cyclic queue or multiprogramming system in which the number of programs J is unlimited. We shall assume, as is realistic, that the CPU service rate outstrips that of the DTU, i.e. $E[C] < E[D]$. Now Feller ([1], pp. 194-198) shows that W_n has the same distribution as the maximum of the partial sums of the unrestricted random walk

$$W_n \overset{(d)}{=} M_n = \max[0, S_1, S_2, \ldots, S_n] \tag{3.2}$$

where

$$S_n = X_1 + X_2 + \ldots + X_n \tag{3.3}$$

and

$$X_k = C_k - D_k. \tag{3.4}$$

To study M_n, invoke Wald's Identity see Feller ([1], p. 603) or Kingman [4]:

$$E\left\{\frac{e^{sS_N}}{[\psi(s)]^N}\right\} = 1, \tag{3.5}$$

N being the random time at which a boundary is reached, and

$$\psi(s) = E[e^{sX}] = E[e^{sC}]\, E[e^{-sD}]. \tag{3.6}$$

Now place a boundary at $x > 0$, and another at $-b$, $b > 0$. Then

$$E\left\{\frac{e^{sS_N}}{\psi^N} \;\middle|\; S_N > x\right\} P\{S_N > x\} + E\left\{\frac{e^{sS_N}}{\psi^N} \;\middle|\; S_N < -b\right\} P\{S_N < -b\} = 1 \tag{3.7}$$

If $E[C] < E[D]$ it may be shown that the equation

$$\psi(s) = 1 \tag{3.8}$$

has a solution at $s = 0$, and one at $\underline{s} > 0$. Put $s = \underline{s}$, let $b \to \infty$, and observe that then

$$P\{S_N > x\} = \frac{1}{E\{e^{\underline{s}S_N} | S_N > x\}} \tag{3.9}$$

This is the probability that the unrestricted random walk S ever exceeds the boundary at x, and is, by (3.2), equal to the probability that the waiting time exceeds x. We write this as

$$P\{W > x\} = \frac{e^{-\underline{s}x}}{E\{e^{\underline{s}(S_N - x)} | S_N > x\}} \tag{3.10}$$

where $S_N - x > 0$ represents the excess; if we neglect the latter we obtain the estimate

$$P\{W > x\} \leq e^{-\underline{s}x};$$

if x is large we have the approximation

$$P\{W > x\} \approx C\, e^{-\underline{s}x}.$$

By the result of Haji and Newell [3], the number, Q, of customers in the queue is the number that arrive during the waiting time of an arbitrary customer; reference is to the stationary distributions of both W and Q. Conditionally,

$$P\{Q \geq n \mid W = x\} = G^{n*}(x), \tag{3.11}$$

where G is the distribution function of D, and $*$ represents Stieltjes convolution. Then, by (3.11) above,

$$P\{Q \geq n\} \approx C \int_0^\infty G^{n*}(x) e^{-\underline{s}x} \, \underline{s} \, dx = C[\hat{G}(\underline{s})]^n \tag{3.12}$$

where $\hat{G}(\underline{s})$ is the Laplace-Stieltjes transform of G, evaluated at $\underline{s}$. This effectively states that, at least under heavy traffic conditions ($\rho = \frac{E[C]}{E[D]}$ barely < 1) the stationary distribution of the number in the system is exponential, but with parameter somewhat different from that of the diffusion approximation:

$$\text{Diffusion:} \quad P\{Q \geq x\} \approx e^{\frac{2\mu}{\sigma^2} x}, \tag{3.13 a}$$

where $\mu = \frac{1}{E[D]} - \frac{1}{E[C]}$ and $\sigma^2 = \frac{Var[D]}{(E[D])} + \frac{Var[C]}{(E[C])}$

$$\text{Wald:} \quad P\{Q \geq x\} \approx e^{[\ell n \, \hat{G}(\underline{s})]x} = [\hat{G}(\underline{s})]^x \tag{3.13 b}$$

see Gaver and Shedler [2]. For a new approximation we then merely replace the ratio $\frac{2\mu}{\sigma^2}$ by $\ell n \, \hat{G}(\underline{s})$ and fit constants as was done in [2]. The relation between the parameters in the diffusion approximation expressed by (3.13a) and that in the approximation resulting from Wald's Identity (3.13b), is considered in the Appendix.

Given the values of μ and σ^2 the stationary diffusion approximation for the distribution F of Q satisfies

$$0 = -\frac{dF}{dx} + \frac{\sigma^2}{2\mu} \frac{d^2F}{dx^2}, \tag{3.14}$$

in which we now propose to replace $\frac{2\mu}{\sigma^2}$ by $\ell n\, \hat{G}(\underline{s})$. We also must determine the constant B in the solution to (3.14).

$$F(x;J) = \frac{1 - B\, e^{-\alpha x}}{1 - B\, e^{-\alpha J}} \tag{3.15}$$

where the latter expression satisfies the boundary condition at $x = J$: $F(J;J) = 1$. Here we have introduced the notation $F(x;J)$ to emphasize the dependence upon the parameter J. The constant $\alpha > 0$ can be determined either by an argument based on asymptotic normality in conflicting renewal processes (see [2]), or as we have argued, using Wald-Haji-Newell results.

We now present two ways in which B can be determined.

4. Fitting the Constant B: Approximations for CPU Utilization.

We suggest and investigate two ways in which the constant B in (3.15) can readily be determined.

Method 1. If $J = \infty$ then it is well known (see Takacs [6], p.142) that server (CPU) utilization is

$$1 - F(0+;\infty) = \rho \equiv \frac{E[C]}{E[D]} \tag{4.1}$$

Hence it follows that to achieve this approximately for large J we should put $B = B_1 = \rho$, from which it follows that

$$F(x;J) = \frac{1 - \rho\, e^{-\alpha x}}{1 - \rho\, e^{-\alpha J}} \qquad 0 \le x \le J; \quad \alpha > 0 \tag{4.2}$$

This approach was taken in [2] with good results for exponential CPU service.

Method 2. With probability $F(J-1;J)$ there is at least one program in residence at the DTU. Hence the long-run input to the CPU should be $1 \cdot F(J-1;J)$, assuming that $E[D] = 1$. Now the long-run output rate from the CPU must equal the input rate, and the output rate approximates $[1 - F(0+;J)]\,\frac{1}{E[C]}$. By this conservation principle, then,

$$\left(\frac{1 - B\, e^{-\alpha(J-1)}}{1 - B\, e^{-\alpha J}}\right) = \frac{B}{E[C]} \left(\frac{1 - e^{-\alpha J}}{1 - B\, e^{-\alpha J}}\right)$$

from which we find that

$$B_2 = \frac{\rho}{1 + \rho\, e^{-\alpha(J-1)} - e^{-\alpha J}}$$

Of course $B_2 \to B_1$ as $J \to \infty$.

We shall shortly provide some numerical comparisons that illustrate the behavior of the two methods when they are applied to actual measurement data.

5. Special Cases.

We now describe the manner in which our approximations may be applied when certain specific distributions are in force.

Case 1: CPU service exponential, $E[C] = \lambda^{-1}$; DTU service Erlang - k, $E[D] = 1$.

In this case equation (3.8) has the form

$$\psi(s) \equiv \left(\frac{\lambda}{\lambda - s}\right)\left(\frac{1}{1 + \frac{s}{k}}\right)^k = 1. \tag{5.1}$$

It must be solved numerically for s, a task that can be carried out by Newton-Raphson iteration.

Case 2: CPU services exponential; DTU service constant, $E[D] = 1$.

For this limiting case of (5.1) let $k \to \infty$ to obtain the equation

$$\left(\frac{\lambda}{\lambda - s}\right)e^{-s} = 1 \tag{5.2}$$

Case 3: CPU services Erlang - k, $E[C] = \lambda^{-1}$; DTU service constant, $E[D] = 1$.

Here we must solve

$$\left(\frac{1}{1 - \frac{s}{\lambda k}}\right)^k e^{-s} = 1. \tag{5.3}$$

Case 4: CPU services hyperexponential; DTU services constant, $E[D] = 1$.

Representation of CPU services by means of a convex combination of exponentials (the hyperexponential distribution) suggests itself according to actual program trace data. This model leads to the equation

$$\left(p \frac{\lambda_1}{\lambda_1 - s} + (1-p) \frac{\lambda_2}{\lambda_2 - s} \right) e^{-s} = 1 \tag{5.4}$$

where

$$E[C] \equiv \lambda^{-1} = p \lambda_1^{-1} + (1-p)\lambda_2^{-1}$$

and p takes on an appropriate value between zero and unity. In practice it is convenient (if not statistically efficient) to fit the parameters of Cases 3 and 4 by the matching of low moments from model and data. Supposing that $\lambda_2^{-1} < E[C] < \lambda_1^{-1}$, it can be shown that, given $E[C]$ and $Var[C]$ such that $\frac{(Var[C])^{1/2}}{E[C]} > 1$, along with λ_2^{-1}, p and λ, are uniquely determined.

Unfortunately, all of the above models require the numerical solution of a transcendental equation in order to generate actual numerical estimates of CPU utilization. This disadvantage is not possessed by the diffusion approximation of [2].

It is of interest that our procedure gives results in complete accord with an exact analysis in one particular case.

Case 5. CPU and DTU services exponential.

This case can easily be analyzed by simple birth-and-death process methods, for which see [2]. Our procedure demands that we first solve

$$\left(\frac{\lambda}{\lambda - s}\right)\left(\frac{1}{1+s}\right) = 1 \tag{5.5}$$

which in this case has the explicit solution $\underline{s} = \lambda - 1$; consequently $\hat{G}(\underline{s}) = \frac{1}{\lambda} \equiv \rho$. Then the approximation yields

$$F(0+;J) = \frac{1 - B_i}{1 - B_i\, \rho^J}. \qquad (i = 1,2)$$

Here B_i refers to the constant B as determined by Method i ($i = 1$ or 2). But for the present model we have

$$B_2 = \frac{\rho}{1 + \rho[\rho^{J-1}] - \rho^J} = \rho = B_1, \tag{5.6}$$

and use of $B_i = \rho$ yields

$$F(0+;J) = \frac{1 - \rho}{1 - \rho^{J+1}} \tag{5.7}$$

so our approximation is in this case equal to the birth-and-death result. For our other cases exact equality will not hold.

6. Numerical Results.

We now present numerical results to indicate the quality of the proposed approximation. Our examples are in the context of a single processor system with two-level memory, multiprogrammed and operated in a demand paging environment. A discussion of cyclic queueing phenomena in such systems is given in Lewis and Shedler [5]. Accordingly, we interpret the CPU service times in our model as execution intervals, i.e. times between page exceptions as programs execute in (constrained) memory of given capacity. We concentrate on Case 4 above (CPU services hyperexponential, DTU services constant) on the basis of our experience that execution intervals often fit well to a hyperexponential model. The assumption of constant DTU service times arises from the consideration of average access time along with the time to transfer a page of information.

In all cases we shall consider, values for p, μ_1, and μ_2 in the hyperexponential were obtained by matching first and second moments of the empirical distribution obtained from actual program data.

Tables 1 and 2 contain numerical results for CPU utilization obtained by the approximation technique (for both methods of fitting the constant B) along with results of exact analysis based on semi-Markov (S-M) methods as given in [5].

Table 1: CPU Utilization Comparisons

J	S-M	Approx B_1	Approx B_2
2	.3903	.1909	.3972
3	.4054	.2486	.4274
4	.4178	.2924	.4440
5	.4280	.3264	.4545
6	.4367	.3534	.4616
7	.4439	.3751	.4668
8	.4501	.3927	.4708
9	.4553	.4072	.4736
10	.4598	.4193	.4759

$E[C] = 4871$, $Var[C[= .26492 \times 10^9$, $\lambda_2^{-1} = 1929$

$E[D] = 10,000$

J	S-M	Approx B_1	Approx B_2
2	.2216	.1455	.2216
3	.2286	.1789	.2313
4	.2333	.2003	.2361
5	.2366	.2144	.2388
6	.2388	.2238	.2404
7	.2403	.2301	.2415
8	.2413	.2344	.2422
9	.2420	.2373	.2426
10	.2425	.2393	.2429

$E[C] = 4871$, $Var[C] = .26492 \times 10^9$, $\lambda_2^{-1} = 1929$

$E[D] = 20,000$

Table 2: CPU Utilization Comparisons

J	S-M	Approx B_1	Approx B_2
2	.4076	.0770	.4249
3	.4281	.1094	.4579
4	.4449	.1385	.4764
5	.4587	.1649	.4882
6	.4702	.1887	.4964
7	.4798	.2105	.5024
8	.4879	.2304	.5070
9	.4948	.2485	.5106
10	.5006	.2654	.5136

$E[C] = 10{,}735$, $Var[C] = .12313 \times 10^{10}$, $\lambda_2^{-1} = 2953$

$E[D] = 20{,}000$

J	S-M	Approx B_1	Approx B_2
2	.5316	.2148	.5993
3	.5548	.2884	.6667
4	.5752	.3481	.7064
5	.5935	.3974	.7326
6	.6098	.4388	.7511
7	.6245	.4741	.7650
8	.6379	.5045	.7757
9	.6500	.5309	.7842
10	.6611	.5542	.7911

$E[C] = 17{,}026$, $Var[C] = .39780 \times 10^{10}$, $\lambda_2^{-1} = 3682$

$E[D] = 20{,}000$

Finally, we present some results of CPU utilization obtained by trace-driven simulation of the cyclic queueing system. By this we mean that CPU service times in the model were taken to be the actual sequence of execution intervals derived from a program trace, J copies of this sequence being multiprogrammed. In Table 3, these trace-driven results are displayed, along with values of CPU utilization obtained by the approximation technique.

Table 3: Trace-Driven Simulation
CPU Utilization Comparisons

J	Trace	Approx B_1	Approx B_2
3	.227	.1789	.2313
6	.229	.2238	.2404

$E[C] = 4871, \quad Var[C] = .26492 \times 10^9$

$E[D] = 20,000$

J	Trace	Approx B_1	Approx B_2
3	.419	.1094	.4579
6	.425	.1887	.4964

$E[C] = 10,735, \quad Var[C] = .12313 \times 10^{10}$

$E[D] = 20,000$

J	Trace	Approx B_1	Approx B_2
3	.538	.2884	.6667
6	.546	.4388	.7511

$E[C] = 17,026, \quad Var[C] = .39780 \times 10^{10}$

$E[D] = 20,000$

7. Summary and Conclusions.

This paper presents the results of approximating processor utilization in multiprogrammed computer systems using ideas of diffusion. In particular, the objective is to develop convenient and nearly explicit formulas relating CPU utilization to simple program parameters and to the level of multiprogramming.

Numerical comparisons indicate that a reasonably effective approximation has been obtained when the constant B_2 is utilized. Examples show that for the actual program traces studied our present approximation is superior to that of [2], which assumed exponentially distributed CPU service times. Data from our trace material is far more skewed (long-tailed) than that yielded by the exponential. Research continues in an attempt to improve the approximate procedures obtained to date. A promising approach is the iteration of our approximate solutions. Of course, an eventual goal is that of obtaining simple but adequate approximations to properties of somewhat more complex and truly realistic networks of servers.

Appendix

The relation between the parameter in the diffusion approximation of [2], as expressed in (3.13a), and that in the approximation resulting from Wald's Identity, (3.13b), will now be considered. Application of Wald's Identity requires that we find the positive root, $\underline{s}$, of (3.8). Let us expand $\psi(s)$ in Taylor's series:

$$\psi(s) = 1 + s\mu_x + \frac{s^2}{2}\sigma_x^2 + R(s) \tag{A-1}$$

where the remainder is $o(s^2)$, provided that required moments exist. Here

$$\begin{aligned} \mu_x &= E(X) = E[C-D] < 0 \\ \sigma_x^2 &= \mathrm{Var}[X] = \mathrm{Var}[C] + \mathrm{Var}[D]. \end{aligned} \tag{A-2}$$

At $\underline{s}$ we have from (A-1) and (3.13b), after dispensing with the root at $s = 0$,

$$\mu_x + \underline{s}\,\frac{\sigma_x^2}{2} + r(\underline{s}) = 0, \tag{A-3}$$

or

$$\frac{2\mu_x}{\underline{s}\,\sigma_x^2} + 1 + \frac{2}{\sigma_x^2}\left[\frac{1}{\underline{s}}\, r(\underline{s})\right] = 0 \tag{A-4}$$

Therefore, if we consider a sequence of queueing situations in which $\underline{s} \to 0$ and σ_x^2 does not approach zero, the remainder term approaches zero, since $r(s) = 0(s)$. We see then that as $\underline{s} \to 0$,

$$\frac{2\mu_x}{\underline{s}\,\sigma_x^2} \to -1 \tag{A-5}$$

or

$$\underline{s} \sim -\frac{2\mu_x}{\sigma_x^2} \tag{A-6}$$

In the event that $\underline{s} \to 0$ our Wald approximation and the approximation of [2] coincide, as will now be shown. For $\underline{s}$ approaching zero, as will be true in heavy traffic,

$$-\ell n\, \hat{G}(\underline{s}) = \underline{s}\, E[D] + o(\underline{s}) \tag{A-7}$$

Consequently the parameter in the Wald-Haji-Newell approximation becomes in heavy traffic

$$-\underline{s}\, E[D] = 2\,\frac{\mu_x E[D]}{\sigma_x^2} = -2\,\frac{(E[D]-E[C])E[D]}{Var[D] + Var[C]}$$

$$= -2\,\frac{\frac{1}{E[C]} - \frac{1}{E[D]}}{\frac{Var[D]}{(E[D])^2 E[C]} + \frac{Var[C]}{(E[D])^2 E[C]}}$$

$$\sim 2\,\frac{\frac{1}{E[D]} - \frac{1}{E[C]}}{\frac{Var[D]}{(E[D])^3} - \frac{Var[C]}{(E[C])^3}} = 2\,\frac{\mu}{\sigma^2} \tag{A-8}$$

For the specific models introduced earlier in Section 5 it is clearly sufficient to allow the mean CPU service time to approach unity from below in order to force $\underline{s}$ to zero. Consider, for example, Case 3: letting $\frac{1}{\lambda} = E[C]$ increase it is apparent that for every fixed s, $\psi(s)$, the left-hand side of (5.3), increases, and $\underline{s}$ moves continuously towards the origin; when $\frac{1}{\lambda} = 1$ there is a (double) root at $s = 0$. A similar effect occurs when, say, $\lambda_1 \to 0$ in (5.4),

a maneuver that allows $E[C]$ to approach unity. Again $\psi(s)$ is increased for every s, and in the limit there is a double root at $s = 0$. Recall that the region of convergence of the transform $\psi(s)$ is $s < \min(\lambda_1,\lambda_2) = \bar{s},\lambda$ and since $\underline{s} < \bar{s}$ a decrease in either λ_1 or λ_2 eventually sends $\underline{s}$ to zero. Examination of the denominator of (3.10) suggests also that if $\underline{s}$ is near zero the expectation is near unity, thus further justifying the use of our approximation.

REFERENCES

[1] Feller, W., *An Introduction to Probability Theory*, Vol. II, Second Edition, John Wiley, New York, 1971.

[2] Gaver, D. P. and Shedler, G. S., "Multiprogramming System Performance via Diffusion Approximations," IBM Research Report RJ-938, Nov. 11, 1971, also to appear in *Operations Research*.

[3] Haji, R. and Newell, G. F., "A Relation Between Stationary Queue and Waiting Time Distributions," *J. Appl. Prob.*, 8, 3, 617-620, 1971.

[4] Kingman, J. F. C., "A Martingale Inequality in the Theory of Queues," *Proc. Camb. Phil. Soc.*, 60, 359-361, 1964.

[5] Lewis, P. A. W. and Shedler, G. S., "A Cyclic-Queue Model of System Overhead in Multiprogrammed Computer Systems," *J. ACM*, 18, 2, 199-220, 1971.

[6] Takacs, L., *Introduction to the Theory of Queues*, Oxford University Press, New York, 1962.

INITIAL DISTRIBUTION LIST

	No. Copies
Defense Documentation Center (DDC) Cameron Station Alexandria, Virginia 22314	12
Library (Code 0212) Naval Postgraduate School Monterey, California 93940	2
Dean of Research Administration Naval Postgraduate School Monterey, California 93940	1
Library (Code 55) Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	3
Professor D. R. Cox Department of Mathematics Imperial College London, England	1
Professor B. Epstein Israel Institute of Technology Technion City Haifa, Israel	1
Professor P. A. P. Moran Department of Statistics Australian National University P. O. Box 4 Canberra, Australia 2600	1
Professor Daryl Daley Department of Statistics Australian National University P. O. Box 4 Canberra, Australia 2600	1
Marvin Denicoff Office of Naval Research Arlington, Virginia 22217	1

Dr. Thomas Varley
Office of Naval Research
Arlington, Virginia 22217 — 1

Professor U. N. Bhat
Computer Science Center
Institute of Technology
Southern Methodist University
Dallas, Texas 75222 — 1

Dr. Richard Nance
Computer Science Center
Institute of Technology
Southern Methodist University
Dallas, Texas 75222 — 1

Professor John Lehoczky
Statistics Department
Carnegie-Mellon University
Schenley Park
Pittsburgh, Pennsylvania 15213 — 1

Professor Donald McNeil
Statistics Department
Princeton University
Princeton, New Jersey 08540 — 1

Professor Carl Harris
School of Engineering
George Washington University
Washington, D. C. 20006 — 1

Dr. Daniel Heyman
Bell Telephone Inc.
Crawford Corner Road
Holmdel, New Jersey 07733 — 1

Dr. Ronald Wolff
Operations Research Center
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 — 1

Dr. Bruce McDonald
Office of Naval Research
Arlington, Virginia 22217 — 1

Professor Walter Smith
Statistics Department
University of North Carolina
Chapel Hill, North Carolina 27514 — 1

Dr. Glenn Bacon 1
Systems Department
IBM Research
Monterey and Cottle Roads
San Jose, California 95111

Professor Leonard Kleinrock 1
Department of Computer Science
University of California
Los Angeles, California 90024

Dr. Bill Mitchell 1
Department of Management Sciences
California State College
Hayward, California 94542

Dr. Mathew Sobel 1
Administrative Science
Yale University
New Haven, Connecticut 06520

Dr. Peter Denning 1
Princeton University
Princeton, New Jersey

Dr. E. G. Goffman
Computer Science
Penn State University
University Park, Pennsylvania 16802

Professor B. Niebel 1
Industrial Engineering
Penn State University
University Park, Pennsylvania 16802

Dr. Jack McCredie 1
Computer Science
Carnegie-Mellon University
Schenley Park
Pittsburgh, Pennsylvania 15213

Dr. P. A. W. Lewis 1
Dr. Kneale Marshall 1
Dr. R. Butterworth 1
Dr. Paul Milch 1
Department of Operations Research
and Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

Dr. Gerald S. Shedler 1
IBM Research Laboratories
Monterey and Cottle Roads
San Jose, California 95111

Dr. Donald P. Gaver 10
Department of Operations Research
and Administrative Sciences
Naval Postgraduate School
Monterey, California 93940

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Approximate Models for Processor Utilization in Multiprogrammed Computer Systems

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

Donald P. Gaver
Gerald S. Shedler

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1972	30	6

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This paper presents results of an approcimation study of cyclic queueing phenomena that occur in multiprogrammed computer systems. Based on Wald's Identity and using ideas of diffusion, the objective is to develop convenient and nearly explicit formulas relating processor utilization in such systems to simple program parameters and the level of multiprogramming. Some numerical results to indicate the quality of the proposed approximation are given.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computers						
Probability						
Multi-programming						
Queueing						